CLAIMS

What is claimed is:

1. A method, comprising:

examining a first instruction to determine a first destination that the first instruction will write on, the first instruction to be in an instruction window;

examining a second instruction to determine a first source that the second instruction will use and a second destination that the second instruction will write on, the second instruction to enter the instruction window;

setting a written on bit associated with the first instruction to a written on state if the first destination and the second destination will be the same operand;

setting a used bit associated with the first instruction to a used state if the first destination and the first source will be the same operand; and,

determining a priority of the first instruction from the written on and used bits.

2. The method of claim 1, further comprising:

determining a number of times that at least one instruction to enter the instruction window after the first instruction will use the first destination as a source for the at least one instruction to enter the instruction window after the first instruction.

3. The method of claim 1, further comprising:

determining that the first instruction is useless if the first destination and the second destination are the same operand and if no instruction to enter the instruction window after the first instruction and before the second instruction will use the first destination as a source for the at least one instruction to enter the instruction window after the first instruction and before the second instruction.

4. The method of claim 3, further comprising eliminating the first instruction from the instruction window.

5. The method of claim 4, wherein the eliminating of the first instruction is delayed until the second instruction writes on the second destination.

6. The method of claim 3, further comprising recording whether the instruction was useless.

7. The method of claim 6, further comprising predicting whether the instruction will be useless based on the recording whether the instruction was useless in past occurrences.

8. The method of claim 1, further comprising:

determining how many instructions will enter the instruction window after the first instruction and before the second instruction if the first destination and the first source are the same operand; and,

refining the priority of the first instruction based on the how many instructions will enter the instruction window after the first instruction and before the second instruction if the first destination and the first source are the same operand.

9. A processor, comprising:

an instruction window including a plurality of instruction entries including a first instruction entry and a second instruction entry, each instruction entry including an instruction field to be occupied by an instruction, a written on bit, a used bit, and a priority field determined from the written on bit and the used bit;

a fetcher to store a prior instruction in the first instruction entry and to store a subsequent instruction in the second instruction entry, the prior instruction and the subsequent instruction include a source and a destination.

10. The processor of claim 9, wherein the written on bit of the first instruction entry to be set to a written on state if the destination of the prior instruction and the destination of the subsequent instruction are the same operand.

11. The processor of claim 9, wherein the used bit of the first instruction entry to be set to a used state if the source of the subsequent instruction and destination of the prior instruction are the same operand.

12. The processor of claim 11, wherein the used bit of the first instruction entry has a default value to be based on the type of instruction occupying the instruction field.

13. The processor of claim 11, wherein the fetcher to further store one or more instructions that intervene between the prior instruction and the subsequent instruction

in one or more instruction entries of the plurality of instruction entries by including an intervening instruction field in each of the plurality of entries

.

14. The processor of claim 13, wherein when the destination of the prior instruction and the destination of the subsequent instruction are the same operand, the intervening instruction field of the first instruction entry to be set to indicate how many of the one or more instruction entries are occupied by the one or more instructions that intervene between the prior instruction and the subsequent instruction.

15. The processor of claim 9, further comprising:

a used count field of the first instruction entry to determine how many instruction entries other than the first instruction entry are occupied by instructions including a source that is the destination of the prior instruction

16. The processor of claim 9, further comprising the processor being configured to record a priority of at least one instruction.

17. The processor of claim 16, further comprising the processor being further configured to predict the priority of the at least one instruction based on the recorded priority.

18. A computer system, comprising:

a processor including

an instruction window having a plurality of instruction entries including a first instruction entry and a second instruction entry, each instruction entry including an instruction field to be occupied by an instruction, a written on bit, a used bit, and a priority field determined from the written on bit and the used bit;

a memory; and,

a fetcher to fetch instructions from the memory and to store a prior instruction in the first instruction entry and to store a subsequent instruction in the second instruction entry, both the prior instruction and subsequent instruction include a source, a destination, the written on bit of the first instruction entry to be set to a written on state if the destination of the prior instruction and the subsequent instruction are the same operand, and the used bit of the first instruction entry to be set to a used state if the source of the subsequent instruction and the destination of the prior instruction are the same operands.

19. The system of claim 18, wherein the processor further comprises:

a third instruction entry of the plurality of instruction entries wherein the fetcher is to store an instruction that intervenes between the prior instruction and the subsequent instruction in the third instruction entry; and,

an intervening instruction field of the first instruction entry to determine how many of the plurality of instruction entries is occupied by an intervening instruction if the source of the subsequent instruction is the destination of the prior instruction and if no source of the intervening instructions is the destination of the prior instruction.

20. The system of claim 18, wherein the processor further comprises:

a used count field of the first instruction entry to determine how many instruction entries other than the first instruction entry are occupied by instructions including a source that is the destination of the prior instruction.

21. The system of claim 18, further comprising the processor being configured to record a priority of at least one instruction.

22. The system of claim 21, further comprising the processor being further configured to predict the priority of the at least one instruction based on the recorded priority.

23. A circuit, comprising:

a written on logic to determine a written on bit;

a used logic to determine a used bit; and,

a priority logic to determine a priority based on the written on and the used bits, the written on, used and priority logics to be associated with a first instruction entry of a plurality of instruction entries in an instruction window, the plurality of instruction entries to include a second instruction entry, the first instruction entry to be occupied by a prior instruction and the second instruction entry to be occupied by a subsequent instruction, the first and second instructions each include a source and a destination.

24. The circuit of claim 23, further comprising the written on logic to set the written on bit to a written on state if the destination of the subsequent instruction is the same operand as the destination of the prior instruction.

25. The circuit of claim 23, further comprising the used logic to set the used bit to a used state if the source of the subsequent instruction is the destination of the prior instruction.

26. The circuit of claim 23, further comprising the priority logic to set the priority to redundant if the written on bit is set to a written on state and the used bit is set to a not used state.

27. The circuit of claim 23, further comprising:

an intervening instruction logic to determine whether a third instruction entry of the plurality of instruction entries is occupied by an instruction that intervenes between the prior instruction and the subsequent instruction.

28. The circuit of claim 23, further comprising:

a used count logic to determine how many of the plurality of instruction entries are occupied by an instruction that is subsequent to the prior instruction and that has a source that is the same operand as the destination of the prior instruction.

29. The circuit of claim 23, further comprising:

a recording logic to record the priority of at least one instruction.

30. The circuit of claim 23, further comprising a predicting logic to predict the priority of the at least one instruction based on the recorded priority.